

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings of claims in the application:

**Listing of Claims:**

1                   1.       (Original) A method of optimizing a query in a multi-tenant database,  
2       said database having one or more data tables, each table having one or more logical columns  
3       defining data categories and one or more logical rows associated with one or more tenants,  
4       wherein a plurality of tenants have data stored in the data tables, the method comprising:  
5                   generating tenant-level statistics for each of said plurality of tenants for each of  
6       the data tables;  
7                   receiving a SQL query; and  
8                   optimizing the SQL query based on the tenant-level statistics.

1                   2.       (Original) The method of claim 1, wherein each tenant includes one or  
2       more associated users, the method further including:  
3                   generating user-level statistics for each user of each tenant for each of the data  
4       tables; and  
5                   optimizing the SQL query based on the user-level statistics.

1                   3.       (Original) The method of claim 2, wherein the user-level statistics are  
2       stored to a user metadata table.

1                   4.       (Original) The method of claim 2, wherein generating user-level statistics  
2       includes determining a total number of distinct rows for each of said plurality of users.

1                   5.       (Original) The method of claim 4, wherein the total number is an  
2       approximate number based on one or more of a) a number of rows viewable by the user and  
3       users below the user in a role hierarchy, b) a number of rows that are shared by a group to which  
4       the user belongs and c) a number of rows that are manually shared to the user by another user or  
5       group of users.

1                   6.       (Original) The method of claim 2, wherein generating user-level statistics  
2 for a user is performed according to one of a) on a scheduled basis, b) after a predetermined  
3 number of queries by the user, and c) each time an unconstrained query is run by the user.

1                   7.       (Original) The method of claim 1, wherein generating tenant-level  
2 statistics is performed on a periodic basis.

1                   8.       (Original) The method of claim 1, wherein generating includes  
2 determining a total number of distinct rows accessible for each of said plurality of tenants.

1                   9.       (Original) The method of claim 8, wherein the tenant-level statistics are  
2 stored to a tenant metadata table.

1                   10.      (Original) The method of claim 1, wherein at least one column of one of  
2 said tables includes data associated with two or more tenants.

1                   11.      (Original) A multi-tenant database system, comprising:  
2                   a database having one or more data tables, each table having one or more columns  
3 defining data categories and one or more rows associated with one or more tenants, wherein a  
4 plurality of tenants have data stored in the data tables;  
5                   a statistics generating module configured to generate tenant-level statistics for  
6 each tenant for each of the data tables; and  
7                   a query optimization module, configured to optimize a database query based on  
8 the tenant-level statistics.

1                   12.      (Original) The multi-tenant database system of claim 11, wherein each  
2 tenant includes one or more associated users, wherein the statistics generating module is further  
3 configured to generate user-level statistics for each user, and wherein the query optimization  
4 module is further configured to optimize the database query based on the user-level statistics.

1                   13.     (Original) The system of claim 12, further including a memory module,  
2 wherein the statistics generating module stores the user-level statistics to a metadata table in the  
3 memory module.

1                   14.     (Original) The system of claim 12, wherein the statistics generating  
2 module generates user-level statistics by determining a total number of distinct rows for each of  
3 said plurality of users.

1                   15.     (Original) The system of claim 14, wherein the total number is an  
2 approximate number based on one or more of a) a number of rows viewable by the user and  
3 users below the user in a role hierarchy, b) a number of rows that are shared by a group to which  
4 the user belongs and c) a number of rows that are manually shared to the user by another user or  
5 group of users.

1                   16.     (Original) The system of claim 12, wherein the statistics generating  
2 module generates user-level statistics for a user according to one of a) on a scheduled basis, b)  
3 after a predetermined number of queries by the user, and c) each time an unconstrained query is  
4 run by the user.

1                   17.     (Original) The system of claim 11, wherein the statistics generating  
2 module generates tenant-level statistics on a periodic basis.

1                   18.     (Original) The system of claim 11, wherein the statistics generating  
2 module generates tenant-level statistics by determining a total number of distinct rows viewable  
3 for each of said plurality of tenants.

1                   19.     (Original) The system of claim 18, further including a memory module,  
2 wherein the statistics generating module stores the tenant-level statistics to a tenant metadata  
3 table in the memory module.

1                   20.     (Original) The system of claim 11, wherein at least one column of one of  
2     said tables includes data associated with two or more tenants.

1                   21.     (Original) A method of optimizing a query in a multi-tenant database,  
2     said database having one or more data tables, each table having one or more logical columns  
3     defining data categories and one or more logical rows associated with one or more tenants,  
4     wherein a plurality of tenants have data stored in the data tables, and wherein each tenant  
5     includes one or more users, the method comprising:  
6                   processing the data tables so as to determine tenant-level statistics for each of said  
7     plurality of tenants;  
8                   processing the data tables so as to determine user-level statistics for each of said  
9     plurality of user;  
10                  receiving a SQL query; and  
11                  optimizing the SQL query based on one or both of the tenant-level statistics and  
12     the user-level statistics.

1                   22.     (Original) The method of claim 21, further including:  
2                   storing the user-level statistics to a user-level metadata table in a memory module;  
3     and  
4                   storing the tenant-level statistics to a tenant-level metadata table in the memory  
5     module.

1                   23.     (Original) The method of claim 21, wherein determining user-level  
2     statistics includes determining a total number of distinct rows for each of said plurality of users,  
3     and wherein determining tenant-level statistics includes determining a total number of distinct  
4     rows for each of said plurality of tenants.

1                   24.     (Original) The method of claim 21, wherein processing the data tables to  
2     determine tenant-level statistics is performed on a periodic basis.

- 1                   25.     (Original) The method of claim 21, wherein processing the data tables to  
2     determine user-level statistics for a user is performed according to one of a) on a scheduled basis,  
3     b) after a predetermined number of queries by the user, and c) each time an unconstrained query  
4     is run by the user.